

Département d'informatique et de génie logiciel  
**Compression de données**  
**IFT-4003/IFT-7023**

**Notes de cours**  
**Techniques de compression par**  
**dictionnaire**

Édition Hiver 2012

Mohamed Haj Taieb

Local: PLT 2113

Courriel: [mohamed.haj-taieb.1@ulaval.ca](mailto:mohamed.haj-taieb.1@ulaval.ca)

**Faculté des sciences et de génie**  
Département de génie électrique et de  
génie informatique



# Plan de la présentation

---

## □ Implémentation entière:

- Introduction
- Dictionnaire statique
- Codage de digramme
- Dictionnaire adaptatif
- LZ77
- LZ88
- LZW
- Applications

# Introduction

---

## ❑ Méthode de compression statistique

- Source générant des symboles indépendants.
- On décorrèle les données avant de les compresser en utilisant un modèle de corrélation.
- Compression dépend de la validité du modèle.

## ❑ Méthode de compression par dictionnaire

- Exploitation de la structure des données.
- Stockage des motifs redondants dans un dictionnaire.
- Le stockage peut être statique: pas d'effacement de motifs → connaissance à priori.
- Ou dynamique: mise à jour du dictionnaire en fonction des données.

# Idée de base

---

## □ Détection des motifs

- Dans la majorité des données il existe une certaine structure: texte, image, code ...
- Soit on effectue une décorrélation avant d'encoder soit on utilise un dictionnaire.
- Identifier les motifs de symboles les plus fréquents.
- Encodage efficace des motifs fréquents: assignation de code court.
- Encodage par défaut pour le reste.

## □ Cependant

- La technique par dictionnaire ne fonctionne pas pour les données aléatoires.

# Motivation

---

## □ Exemple

- Alphabet: 26 lettres + 6 points de ponctuation → 32.
- Texte à mots de 4 lettres x 5 bits →  $R_1 = 20$  bits/mots
- $32^4 = 2^{20} = 1,048,576$  possibilités de mots.
- On regroupe les 256 mots les plus fréquents dans un dictionnaire → 8 bits + un bit '1' pour indiquer que c'est un mot du dictionnaire.
- Le reste des mots code: 20 bits + un bit '0'.
- Débit:  $R_2 = 9p + 21(1-p) = 21 - 12p \leq R_1 = 20 \rightarrow p \geq 0.084$ .

## □ Remarque

- Si la source est complètement aléatoire on a  $p = 256/220 = 2/11 = 0.00024414 \rightarrow R_2 = 20.997 > R_1 = 20$ .

# Dictionnaire statique

---

## □ Compression par dictionnaire statique

- Connaissance à priori de la source d'information.
- Design d'un dictionnaire basé sur les motifs récurrents.
- Technique fonctionne pour des applications spécifiques seulement.
- Sinon ça peut résulter en expansion des données.

## □ Codage digramme

- Dictionnaire statique=alphabet original + un nombre de paires de lettres ou digrammes les plus fréquents.
- Exemple: dictionnaire de taille = 256 pour les 95 caractères ASCII imprimables. → 161 entrées restantes.
- 161 entrées pour les digrammes les plus fréquents.

# Exemple: Codage digramme

## □ Exemple: séquence abracadabra

- Alphabet de 5 lettres:  $A = \{a, b, c, d, r\}$ .
- Connaissance à priori de la source → dictionnaire:

Entrée	Code	Entrée	Code
a	000	r	100
b	001	ab	101
c	010	ac	110
d	011	ad	111

## □ Codage

- On lit d'abord une paire et on vérifie si ça se trouve dans le dictionnaire. Sinon on encode la lettre toute seule.

# Exemple: Codage digramme

## □ Exemple: séquence abracadabra

- **ab**racadabra → **101**
- ab**rac**adabra → 101 **X**
- ab**rac**adabra → 101**100**
- ab**rac**adabra → 101100**110**
- abra**cad**abra → 101100110**111**
- abra**cad**abra → 101100110111**101**
- abra**cad**abra → 101100110111101 **X**
- abra**cad**abra → 101100110111101**100**
- abra**cad**abra → 101100110111101100**000**

Entrée	Code
a	000
b	001
c	010
d	011
r	100
ab	101
ac	110
ad	111



# Dictionnaire adaptatif

---

## □ Origines

- Les bases des dictionnaires adaptatif remontent à deux articles historiques de Jacob Ziv et Abraham Lempel publiés en 1977 et 1978.
- 2 approches différentes pour la construction adaptative des dictionnaires:
  - → LZ77 ou LZ1
  - → LZ78 ou LZ2
- Chaque approche a conduit à un nombre de variations.
- → LZW en 1984 par Terry Welch.

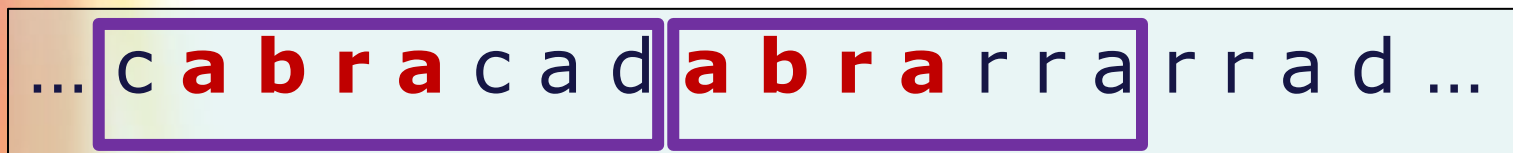
# Approche LZ77 (1)

## □ Dictionnaire

- Le dictionnaire est une portion de la séquence précédemment encodée.
- Utilisation d'une fenêtre glissante.
- Fenêtre comporte deux parties:
  - Bus de recherche: portion précédemment encodé.
  - Bus d'inspection avenir: une portion à encodé.
- Repérer la correspondance maximale entre la séquence à venir et la séquence passée.

Bus de recherche=8

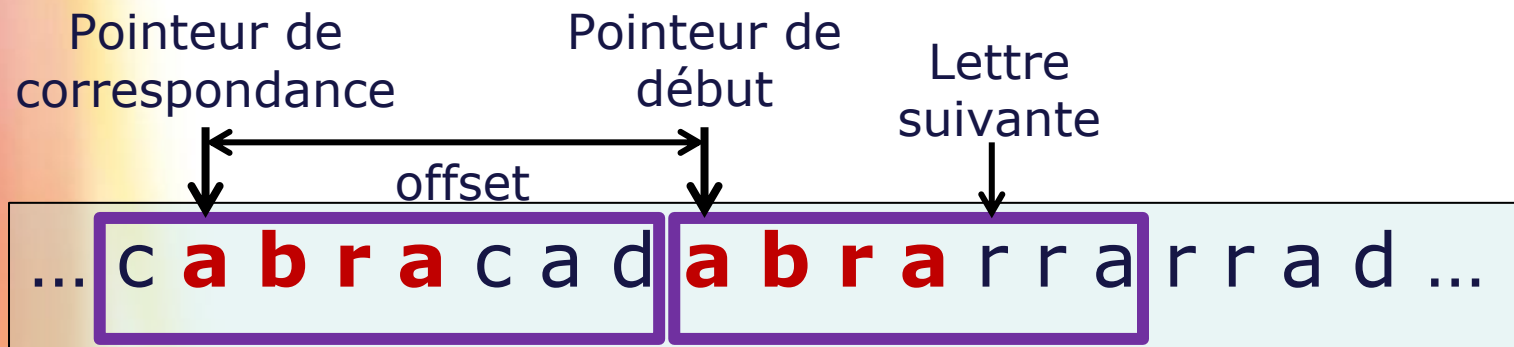
Bus d'inspection=7



# Approche LZ77 (2)

## □ Procédure

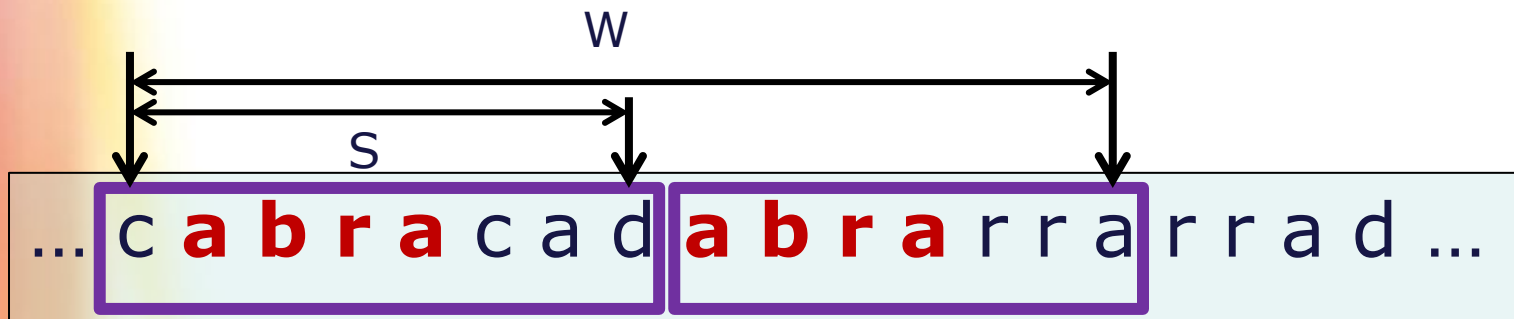
- (O)ffset=Pointeur de début - pointeur de corresp. =7
- (L)ongueur= nombre maximale de symboles consécutifs et similaires =4.
- (C)ode= mot code de la lettre suivante dans le bus d'inspection= r
- Le code est envoyé au cas où il n y a aucune correspondance →  $\langle O=0, L=0, C(r) \rangle$ .



# Approche LZ77 (3)

## □ Encodage avec un codage de longueur fixe

- Taille du bus de recherche =  $S \rightarrow \lceil \log_2 S \rceil$
- Taille de la fenêtre (recherche et inspection) =  $W \rightarrow \lceil \log_2 W \rceil$
- Taille de l'alphabet =  $A \rightarrow \lceil \log_2 A \rceil$
- Encodage (o, l, c)  $\rightarrow \lceil \log_2 S \rceil + \lceil \log_2 W \rceil + \lceil \log_2 A \rceil$
- La longueur de correspondance  $l$  peut dépasser la longueur de la fenêtre de recherche:  $l \rightarrow \lceil \log_2 W \rceil$  et non pas  $\lceil \log_2 S \rceil$ .

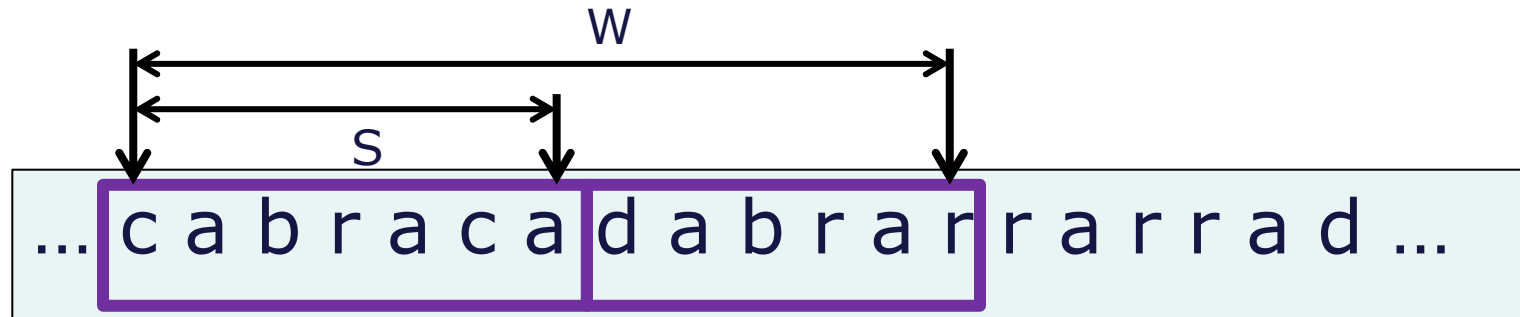


# Exemple: Encodage LZ77 (1)

❑ Séquence à encoder: c a b r a c a d a b r a r r a d

▪  $W=13$

▪  $S=7$



❑ Étape 1: Envoi de  $\langle \text{offset}=0, \text{longueur}=0, \text{Code}(d) \rangle$



▪ Aucune correspondance de d dans le bus de recherche.

$\langle 0, 0, C(d) \rangle$

## Exemple: Encodage LZ77 (2)

❑ Séquence à encoder: c a b r a c a d a b r a r r a r r a d

❑ Étape 2 (a): Envoi de  $\langle \text{offset}=2, \text{longueur}=1, \text{Code}(b) \rangle$   
Envoi de  $\langle \text{offset}=4, \text{longueur}=1, \text{Code}(b) \rangle$

... c a b r a c a d a b r a r r a r r a d ...

❑ Étape 2 (b) : Envoi de  $\langle \text{offset}=7, \text{longueur}=4, \text{Code}(r) \rangle$

... c a b r a c a d a b r a r r a r r a d ...

- → On choisit la longueur de correspondance maximale.

$\langle 0, 0, C(d) \rangle \quad \langle 7, 4, C(r) \rangle$

# Exemple: Encodage LZ77 (3)

❑ Séquence à encoder: c a b r a c a d a b r a r r a d

❑ Étape 3 (a): Envoi de  $\langle \text{offset}=1, \text{longueur}=1, \text{Code}(a) \rangle$

... c a b r a c a d a b r a r r a d ...

❑ Étape 3 (b): Envoi de  $\langle \text{offset}=3, \text{longueur}=3, \text{Code}(r) \rangle$

... c a b r a c a d a b r a r r a d ...

❑ Étape 3 (c): Envoi de  $\langle \text{offset}=3, \text{longueur}=5, \text{Code}(d) \rangle$

... c a b r a c a d a b r a r r a d ...

$\langle 0, 0, C(d) \rangle \langle 7, 4, C(r) \rangle \langle 3, 5, C(d) \rangle$

# Exemple: Décodage LZ77

❑ Séquence à décoder:  $\langle 0, 0, C(d) \rangle \langle 7, 4, C(r) \rangle \langle 3, 5, C(d) \rangle$

❑ Étape 1:  $\langle 0, 0, C(d) \rangle$

... c a b r a c a d

❑ Étape 2:  $\langle 7, 4, C(r) \rangle$

... c a b r a c a d a b r a r

❑ Étape 3 (a):  $\langle 3, 5, C(d) \rangle$  le bus s'arrête à  $l=3$ .

... c a b r a c a d a b r a r r a r

❑ Étape 3 (b): On a pris 3 lettres il reste 2 autres puis d

... c a b r a c a d a b r a r r a r r a d



# Variantes de LZ77

---

## □ LZ77:

- Pas besoin d'aucune connaissance à priori sur la source.
- Asymptotiquement les performances s'approchent du cas de connaissance totale de toutes les statistiques.
- Hypothèse: les motifs se répètent localement.
- Cette hypothèse est écartée dans l'approche LZ78.

## □ Amélioration de LZ77:

- Encodage de  $\langle o, l, c \rangle$  avec un code à longueur variable.
- Variation de la taille des 2 bus.
- Élimination du triple  $\langle o, l=0, c \rangle \rightarrow$  utilisation d'un bit drapeau pour indiquer si c'est un symbole seul.
- $\rightarrow$  envoi de la paire  $\langle o, l \rangle$  seulement  $\rightarrow$  Variante LZSS.

# Approche LZ78

---

## □ LZ77:

- LZ77 assume implicitement une redondance locale.
- Supposez une séquence périodique  $T=9$  et  $S=8$ .
- $\rightarrow$  expansion. Si  $S=9 \rightarrow$  compression efficace.

## □ LZ78:

- Élimination du buffer  $\rightarrow$  utilisation d'un dictionnaire.
- Construction du dictionnaire à l'encodeur et au décodeur de façon identique.
- $\langle$ (i)ndice dans le dictionnaire assurant une similitude maximale avec l'entrée , (c)ode de la lettre suivante $\rangle$ .
- Pas de correspondance  $\rightarrow$  Indice '0'.
- $\langle i$ , ancienne séquence+c $\rangle$  ajoutée au dictionnaire.

# Exemple: approche LZ78 (1)

- Séquence: wabba□wabba□wabba□wabba□woo□woo□woo
  - □ désigne l'espace.

Résultat d'encodage	Dictionnaire Indice	Entrée
$\langle 0, c(w) \rangle$	1	w

# Exemple: approche LZ78 (2)

- Séquence: w**a**bbba□wabba□wabba□wabba□woo□woo□woo
  - □ désigne l'espace.

<b>Résultat d'encodage</b>	<b>Dictionnaire</b>	
	<b>Indice</b>	<b>Entrée</b>
$\langle 0, c(w) \rangle$	1	w
$\langle 0, c(a) \rangle$	2	a

# Exemple: approche LZ78 (3)

- Séquence: wa**b**a□wabba□wabba□wabba□woo□woo□woo
  - □ désigne l'espace.

Résultat d'encodage	Dictionnaire Indice	Entrée
$\langle 0, c(w) \rangle$	1	w
$\langle 0, c(a) \rangle$	2	a
$\langle 0, c(b) \rangle$	3	b

# Exemple: approche LZ78 (4)

- Séquence: wab**ba** □ wabba □ wabba □ wabba □ woo □ woo □ woo
  - □ désigne l'espace.

<b>Résultat d'encodage</b>	<b>Dictionnaire</b>	
	<b>Indice</b>	<b>Entrée</b>
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba

# Exemple: approche LZ78 (5)

- Séquence: wabba□wabba□wabba□wabba□woo□woo□woo
  - désigne l'espace.

<b>Résultat d'encodage</b>	<b>Dictionnaire</b>	
	<b>Indice</b>	<b>Entrée</b>
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, □>	5	□

# Exemple: approche LZ78 (6)

- Séquence: wabba □ **wa** bba □ wabba □ wabba □ woo □ woo □ woo
- désigne l'espace.

Résultat d'encodage	Indice	Dictionnaire Entrée
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, □>	5	□
<1, c(a)>	6	wa



# Exemple: approche LZ78 (7)

- Séquence: wabba□wa**bb**a□wabba□wabba□woo□woo□woo
- désigne l'espace.

Résultat d'encodage	Dictionnaire Indice	Dictionnaire Entrée
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, □>	5	□
<1, c(a)>	6	wa
<3, c(b)>	7	bb

# Exemple: approche LZ78 (8)

- Séquence: wabba□wabb**a**□wabba□wabba□woo□woo□woo
  - désigne l'espace.

<b>Résultat d'encodage</b>	<b>Dictionnaire</b>	
	<b>Indice</b>	<b>Entrée</b>
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, c(□)>	5	□
<1, c(a)>	6	wa
<3, c(b)>	7	bb
<2, c(□)>	8	<b>a</b> □

# Exemple: approche LZ78 (9)

- Séquence: wabba□wabba□wabba□wabba□woo□woo□woo
- désigne l'espace.

<b>Résultat d'encodage</b>	<b>Dictionnaire</b>	
	<b>Indice</b>	<b>Entrée</b>
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, c(□)>	5	□
<1, c(a)>	6	wa
<3, c(b)>	7	bb
<2, c(□)>	8	a□
<6, c(b)>	9	wab

# Exemple: approche LZ78 (10)

- Séquence: wabba□wabba□wabba□wabba□woo□woo□woo
- désigne l'espace.

Résultat d'encodage	Dictionnaire Indice	Dictionnaire Entrée
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, c(□)>	5	□
<1, c(a)>	6	wa
<3, c(b)>	7	bb
<2, c(□)>	8	a□
<6, c(b)>	9	wab

Résultat d'encodage	Dictionnaire Indice	Dictionnaire Entrée
<4, c(□)>	10	ba□

# Exemple: approche LZ78 (11)

- Séquence: wabba□wabba□wabba□wabbawoo□woo□woo
- désigne l'espace.

Résultat d'encodage	Dictionnaire Indice	Entrée
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, c(□)>	5	□
<1, c(a)>	6	wa
<3, c(b)>	7	bb
<2, c(□)>	8	a□
<6, c(b)>	9	wab

Résultat d'encodage	Dictionnaire Indice	Entrée
<4, c(□)>	10	ba□
<9, c(b)>	11	wabb

# Exemple: approche LZ78 (12)

- Séquence: wabba□wabba□wabba□wabbawoo□woo□woo
- □ désigne l'espace.

Résultat d'encodage	Dictionnaire Indice	Dictionnaire Entrée
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, c(□)>	5	□
<1, c(a)>	6	wa
<3, c(b)>	7	bb
<2, c(□)>	8	a□
<6, c(b)>	9	wab

Résultat d'encodage	Dictionnaire Indice	Dictionnaire Entrée
<4, c(□)>	10	ba□
<9, c(b)>	11	wabb
<8, c(w)>	12	a□w

# Exemple: approche LZ78 (13)

- ❑ Séquence: wabba❑wabba❑wabba❑wabba❑w**o**o❑w**o**o❑w**o**o
  - ❑ désigne l'espace.

Résultat d'encodage	Dictionnaire Indice	Dictionnaire Entrée
<0, c(w)>	1	w
<0, c(a)>	2	a
<0, c(b)>	3	b
<3, c(a)>	4	ba
<0, c(❑)>	5	❑
<1, c(a)>	6	wa
<3, c(b)>	7	bb
<2, c(❑)>	8	a❑
<6, c(b)>	9	wab

Résultat d'encodage	Dictionnaire Indice	Dictionnaire Entrée
<4, c(❑)>	10	ba❑
<9, c(b)>	11	wabb
<8, c(w)>	12	a❑w
<0, c(o)>	13	o
<13, c(❑)>	14	o❑
<1, c(o)>	15	wo
<14, c(w)>	16	o❑w
<13, c(o)>	17	oo

**Il faut stopper la croissance du dictionnaire à un certain point**

# Variation du LZ78 → LZW

---

## □ Séquence:

- Terry Welch: pas nécessaire d'encoder le second élément de  $\langle i, c \rangle$ . Il suffit juste d'envoyer l'indice  $i$ .
- Pour cette raison le dictionnaire doit contiendra toutes les lettres de l'alphabet.
- Les entrées à l'encodeur sont accumulées dans un motif  $p$  se trouvant dans le dictionnaire.
- Lors de la réception d'une lettre 'a' tq  $p^*a$  ne figure pas dans le dictionnaire:
  1. Transmission de l'indice de  $p$
  2.  $p^*a$  insérée dans le dictionnaire
  3. On commence un nouveau pattern avec la lettre 'a'.



# Exemple: Encodeur LZW (1)

- ❑ Séquence: wabba□wabba□wabba□wabba□woo□woo□woo
  - Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w

❑ Code:

# Exemple: Encodeur LZW (2)

❑ Séquence: wabba□wabba□wabba□wabba□woo□woo□woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa

❑ Code: 5

# Exemple: Encodeur LZW (3)

❑ Séquence: w**ab**ba□wabba□wabba□wabba□woo□woo□woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab

❑ Code: 5 2

# Exemple: Encodeur LZW (4)

□ Séquence: wa**bb**a□wabba□wabba□wabba□woo□woo□woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb

□ Code: 5 2 3

# Exemple: Encodeur LZW (5)

□ Séquence: wab**ba** □ wabba □ wabba □ wabba □ woo □ woo □ woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

□ Code: 5 2 3 3

# Exemple: Encodeur LZW (6)

❑ Séquence: wabb**a** □ wabba □ wabba □ wabb**a** □ woo □ woo □ woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée	Dictionnaire Indice	Entrée
1	□	10	a□
2	a	11	□w
3	b	12	wab
4	o	13	bba
5	w	14	a□w
6	wa	15	wabb
7	ab	16	ba□
8	bb	17	□wa
9	ba	18	abb

❑ Code: 5 2 3 3 **2 1 6 8 10 12 9 11 7**

# Exemple: Encodeur LZW (7)

❑ Séquence: wabba □ wabba □ wabba □ wabba □ wabba □ woow □ woow □ woow

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16

# Exemple: Encodeur LZW (8)

❑ Séquence: wabba □ wabba □ wabba □ wabba □ **w**o □ woo □ woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w
20	wo

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5



# Exemple: Encodeur LZW (9)

❑ Séquence: wabba□wabba□wabba□wabba□w**oo**□woo□woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w
20	wo
21	oo

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4

# Exemple: Encodeur LZW (10)

❑ Séquence: wabba □ wabba □ wabba □ wabba □ wo **o** □ woo □ woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w
20	wo
21	oo
22	o□

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4

# Exemple: Encodeur LZW (11)

❑ Séquence: wabba □ wabba □ wabba □ wabba □ woo □ **wo** □ woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w
20	wo
21	oo
22	o□
23	□wo

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 **11**

# Exemple: Encodeur LZW (12)

□ Séquence: wabba □ wabba □ wabba □ wabba □ woo □ w **oo** □ woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w
20	wo
21	oo
22	o□
23	□wo
24	oo□

□ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 **21**

# Exemple: Encodeur LZW (13)

❑ Séquence: wabba□wabba□wabba□wabba□woo□woo□woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w
20	wo
21	oo
22	o□
23	□wo
24	oo□
25	□woo

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23

# Exemple: Encodeur LZW (14)

□ Séquence: wabba□wabba□wabba□wabba□woo□woo□woo

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

Dictionnaire Indice	Entrée
10	a□
11	□w
12	wab
13	bba
14	a□w
15	wabb
16	ba□
17	□wa
18	abb

Dictionnaire Indice	Entrée
19	ba□w
20	wo
21	oo
22	o□
23	□wo
24	oo□
25	□woo

□ Code: 5 2 3 3 2 1 6 8 10 12 9 7 9 11 7 16 5 4 4 11 21 23 4

# Exemple: Décodage LZW (1)

□ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	w*?
7	
8	
9	

On va imiter l'encodeur lors de la création du dictionnaire.

□ Séquence: w

# Exemple: Décodage LZW (2)

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	$\square$
2	a
3	b
4	o
5	w
6	wa
7	a*?
8	
9	

❑ Séquence: w a



# Exemple: Décodage LZW (3)

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 7 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée
1	□
2	a
3	b
4	o
5	w
6	wa
7	ab
8	bb
9	ba

❑ Séquence: w a b b a

# Exemple: Décodage LZW (4)

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée	Dictionnaire Indice	Entrée
1	$\square$	10	a $\square$
2	a	11	$\square$ *?
3	b		
4	o		
5	w		
6	wa		
7	ab		
8	bb		
9	ba		

❑ Séquence: w a b b a  $\square$

# Exemple: Décodage LZW (5)

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée	Dictionnaire Indice	Entrée
1	$\square$	10	a $\square$
2	a	11	$\square$ w
3	b	12	wa*?
4	o		
5	w		
6	wa		
7	ab		
8	bb		
9	ba		

❑ Séquence: w a b b a  $\square$  w a

# Exemple: Décodage LZW (6)

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée	Dictionnaire Indice	Entrée
1	$\square$	10	a $\square$
2	a	11	$\square$ w
3	b	12	wab
4	o	13	bb*?
5	w		
6	wa		
7	ab		
8	bb		
9	ba		

❑ Séquence: w a b b a  $\square$  w a bb

# Exemple: Décodage LZW (6)

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée	Dictionnaire Indice	Entrée
1	$\square$	10	a $\square$
2	a	11	$\square$ w
3	b	12	wab
4	o	13	bba
5	w	14	a $\square$ *?
6	wa		
7	ab		
8	bb		
9	ba		

❑ Séquence: w a b b a  $\square$  wa bb a $\square$

# Exemple: Décodage LZW (7)

□ Code: 5 2 3 3 2 1 6 8 10 **12** 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée	Dictionnaire Indice	Entrée
1	□	10	a□
2	a	11	□w
3	b	12	wab
4	o	13	bba
5	w	14	a□w
6	wa	15	wab*?
7	ab		
8	bb		
9	ba		

□ Séquence: w a b b a □ wa bb a□ wab

# Exemple: Décodage LZW (7)

❑ Code: 5 2 3 3 2 1 6 8 10 12 9 11 7 16 5 4 4 11 21 23 4

- Alphabet  $A = \{\square, a, b, o, w\}$ .

Dictionnaire Indice	Entrée	Dictionnaire Indice	Entrée	Indice	Entrée
1	$\square$	10	a $\square$	19	ba $\square$ w
2	a	11	$\square$ w	20	wo
3	b	12	wab	21	oo
4	o	13	bba	22	o $\square$
5	w	14	a $\square$ w	23	$\square$ wo
6	wa	15	wabb	24	oo $\square$
7	ab	16	ba $\square$	25	$\square$ woo
8	bb	17	$\square$ wa		
9	ba	18	abb		

❑ Séquence: w a b b a  $\square$  wa bb a $\square$  wab ba  $\square$ w ab ba $\square$  w o o  $\square$ w oo  $\square$ wo o

# Exemple: Cas spécial de LZW (1)

---

□ Séquence: a**b**abababababa...

- Alphabet  $A = \{a, b\}$ .

<b>Dictionnaire Indice</b>	<b>Entrée</b>
1	a
2	b
3	ab

□ Code: 1



# Exemple: Cas spécial de LZW (2)

□ Séquence: a**b**abababababa...

- Alphabet  $A = \{a, b\}$ .

<b>Dictionnaire Indice</b>	<b>Entrée</b>
1	a
2	b
3	ab
4	ba

□ Code: 1 2

# Exemple: Cas spécial de LZW (3)

□ Séquence: ababababababa...

- Alphabet  $A = \{a, b\}$ .

<b>Dictionnaire Indice</b>	<b>Entrée</b>
1	a
2	b
3	ab
4	ba
5	aba

□ Code: 1 2 3

# Exemple: Cas spécial de LZW (4)

□ Séquence: ababababababa...

- Alphabet  $A = \{a, b\}$ .

<b>Dictionnaire Indice</b>	<b>Entrée</b>
1	a
2	b
3	ab
4	ba
5	aba
6	abab

□ Code: 1 2 3 5

# Exemple: Cas spécial de LZW (5)

□ Séquence: ababababababa...

- Alphabet  $A = \{a, b\}$ .

<b>Dictionnaire Indice</b>	<b>Entrée</b>
1	a
2	b
3	ab
4	ba
5	aba
6	abab
7	bab

□ Code: 1 2 3 5 4

# Exemple: Cas spécial décodage LZW (1)

□ Code: 1 2 3 5 4

- Alphabet  $A = \{a, b\}$ .

Dictionnaire Indice	Entrée
1	a
2	b
3	a*?

□ Séquence: a

# Exemple: Cas spécial décodage LZW (2)

□ Code: 1 **2** 3 5 4

- Alphabet  $A = \{a, b\}$ .

Dictionnaire Indice	Entrée
1	a
2	b
3	ab
4	b*?

□ Séquence: a b

# Exemple: Cas spécial décodage LZW (3)

□ Code: 1 2 **3** 5 4

- Alphabet  $A = \{a, b\}$ .

Dictionnaire Indice	Entrée
1	a
2	b
3	ab
4	ba
5	ab*?

□ Séquence: a b ab

# Exemple: Cas spécial décodage LZW (4)

□ Code: 1 2 3 **5** 4

- Alphabet  $A = \{a, b\}$ .

Dictionnaire Indice	Entrée
1	a
2	b
3	ab
4	ba
5	ab*?

ab\*? → Inconnu

Indice	Entrée
1	a
2	b
3	ab
4	ba
5	aba

□ Séquence: a b ab bab **ab\*?**

Ainsi on peut savoir c'est quoi la lettre suivante: c'est 'a'. On complète alors l'entrée 5



# Applications: commande Compress de UNIX

---

## □ Compression de fichier avec la technique LZW

- Commande compress d'UNIX.
- Dictionnaire adaptative: au départ de taille  $512=2^9$ .
- Lorsque les 512 entrées du dictionnaires sont remplies, une extension à 1024 est effectuée.
  - Ajouts de 0 aux 512 anciennes entrées
  - Longueur du mots code devient 10.
  - Taille maximale  $2^{b_{\max}}$  avec  $b_{\max}$  entre 9 et 16.

## □ Remplissage des $2^{b_{\max}}$ entrées du dictionnaire

- Calcul du taux de compression C.
- Si  $C > \text{Seuil}$  → dictionnaire statique
- Sinon → destruction du dictionnaire: on recommence.

# Applications: Graphics Interchange Format (GIF)

---

## ❑ Compression d'image avec la technique LZW

- Très similaire à la commande compress d'UNIX.
- Clear code: remise à l'état initial. Taille=256=2<sup>8</sup>.
- Extension jusqu'à 2<sup>12</sup>=4096 → dictionnaire statique.

## ❑ Format

- Mots code stockés dans des blocs de taille < 255.
- Chaque bloc a une entête indiquant la taille du bloc.
- Mot code de terminaison: 8 bits '0'.
- Fin de compression indiqué par une valeur 2<sup>8+1</sup>.

## ❑ Efficacité de LZW dans la compression d'image

- On s'attend pas à une grande efficacité car dans une image il n'y a pas beaucoup de motifs répétitives.

# Applications: Graphics Interchange Format (GIF)

Image	GIF	Arithmetic Coding of Pixel Values	Arithmetic Coding of Pixel Differences
Sena	51,085	53,431	31,847
Sensin	60,649	58,306	37,126
Earth	34,276	38,248	32,137
Omaha	61,580	56,061	51,393

- Pour l'image earth qui contient des régions constantes → efficacité de l'approche par dictionnaire.
- Pour les autres images il faut effectuer un prétraitement pour avoir des séquences qui se prêtent mieux au codage par dictionnaire → PNG.

# Applications: Portable Network Graphics (PNG) [1]

---

## □ Basé sur l'algorithme deflate:

- Compression par dictionnaire utilisant les codes de Huffman.
- A chaque itération lecture de 3 octets.
- Recherche d'une correspondance pour ces 3 octets en déplaçant un offset de 1 à 32768 (soit une zone de recherche de 4096 pixels).
- Si aucune correspondance translation de la fenêtre par 1 octet → envoi littéralement de l'octet (appelé aussi littéral).
- Sinon envoi de la paire <longueur de la correspondance, offset>. [longueur=3 .. 258]

# Applications: Portable Network Graphics (PNG) [2]

---

## □ Encodage du littéral et de la longueur de correspondance:

- Alphabet de taille 286: 0—285.
- Littéral= octet → 0—255.
- Indice 256: symbole fin de bloc.
- Il reste 29 indice de 257—285 pour encoder la longueur de correspondance qui varie de 3—258.
- Il faut rajouter des bits de sélection pour pouvoir couvrir toutes les valeur de longueur de correspondance.
- Les indices sont encodés par un code de Huffman.
- L'offset (1—32786) est encodé en subdivisant les valeurs possibles en 30 plages encodé avec Huffman.
- Dans chaque plage on utilise des bits de sélection.

# Applications: Portable Network Graphics (PNG) [3]

## Encodage de la longueur de correspondance:

Index	# of selector bits	Length	Index	# of selector bits	Length	Index	# of selector bits	Length
257	0	3	267	1	15,16	277	4	67-82
258	0	4	268	1	17,18	278	4	83-98
259	0	5	269	2	19-22	279	4	99-114
260	0	6	270	2	23-26	280	4	115-130
261	0	7	271	2	27-30	281	5	131-162
262	0	8	272	2	31-34	282	5	163-194
263	0	9	273	3	35-42	283	5	195-226
264	0	10	274	3	43-50	284	5	227-257
265	1	11, 12	275	3	51-58	285	0	258
266	1	13, 14	276	3	59-66			

# Applications: Portable Network Graphics (PNG) [4]

## ❑ Encodage de l'indice avec Huffman:

Index Ranges	# of bits	Binary Codes
0–143	8	00110000 through 10111111
144–255	9	110010000 through 111111111
256–279	7	0000000 through 0010111
280–287	8	11000000 through 11000111

# Applications: Filtre de compression du PNG

---

## ❑ Sens de balayage de l'image

- La lecture de l'image se fait ligne par ligne de gauche à droite et de haut en bas.
- On applique des filtre sur une base de ligne par ligne.

## ❑ Type de filtre

- Sub: différence avec le pixel précédent
- Up: différence avec le pixel au-dessus.
- Moyenne: différence avec la moyenne des pixels de gauche et de dessus.
- Le plus proche: choix du pixel le plus proche parmi les 3 pixels suivants: pixel à gauche, pixel au-dessus et pixel au dessus à gauche.



# Applications: Performance du PNG

- ❑ PNG est plus performant que le GIF pour la compression d'image:

Image	PNG	GIF	Arithmetic Coding of Pixel Values	Arithmetic Coding of Pixel Differences
Sena	31,577	51,085	53,431	31,847
Sensin	34,488	60,649	58,306	37,126
Earth	26,995	34,276	38,248	32,137
Omaha	50,185	61,580	56,061	51,393